

**Amendments to the Claims**

Please cancel claims 1 and 7. Please amend claims 2, 4, 8 and 9. The currently pending claims after amendment are listed below.

1. (Cancelled)

2. (Currently Amended) The method for database query optimization of ~~claim 1~~ + claim 5,  
wherein said respective processor resource assignment of each partition comprises a respective  
number of virtual processors of each partition, said respective number being an integer.  
3

1 3. (Original) The method for database query optimization of claim 2, wherein said step of  
2 defining a plurality of logical partitions comprises defining at least one set of processors which is  
3 shared by a set of said logical partitions, said set of said logical partitions containing at least two  
4 partitions, said respective processor resource assignment of each partition of said set of partitions  
5 including said set of processors.

1 4. (Currently Amended) The method for database query optimization of ~~claim 1~~ + claim 6,  
2 further comprising the step of:  
3 saving said first search strategy in a persistent object for later execution, said saving step  
4 including saving said first processor resource assignment in said object.

1 5. (Previously Presented) A method for database query optimization in a computer system  
2 having a plurality of central processors, comprising the steps of:  
3 defining a plurality of logical partitions of said computer system, each logical partition  
4 having a respective processor resource assignment, wherein each task executing in said computer

5 system is assigned to a respective one of said logical partitions and wherein the definition of a  
6 plurality of logical partitions may be dynamically altered;  
7 defining a database query;  
8 constructing a first search strategy for said database query, said first search strategy being  
9 dependent on a first processor resource assignment at the time said step of constructing a first  
10 search strategy is performed;  
11 invoking said database query for execution in a first logical partition, said invoking step  
12 being performed after said step of constructing a first search strategy;  
13 comparing a second processor resource assignment to said first processor resource  
14 assignment, said second processor resource assignment being associated with said first logical  
15 partition at the time said invoking said database query for execution step is performed;  
16 automatically constructing a second search strategy dependent on said second processor  
17 resource assignment, said step of automatically constructing a second search strategy being  
18 performed dependent on the results of said comparing step;  
19 saving said first search strategy in a persistent object for later execution, said saving step  
20 including saving said first processor resource assignment in said object;  
21 invoking a previously saved search strategy for execution in a second logical partition, said  
22 second logical partition being different from said first logical partition;  
23 identifying a third processor resource assignment associated with said second logical  
24 partition;  
25 comparing said third processor resource assignment to said first processor resource  
26 assignment; and  
27 automatically constructing a third search strategy for execution of said database query  
28 depending on the results of said comparing step.

1       6. (Previously Presented) A method for database query optimization in a computer system  
2       having a plurality of central processors, comprising the steps of:  
3              defining a plurality of logical partitions of said computer system, each logical partition  
4       having a respective processor resource assignment, wherein each task executing in said computer  
5       system is assigned to a respective one of said logical partitions and wherein the definition of a  
6       plurality of logical partitions may be dynamically altered;  
7              defining a database query;  
8              constructing a first search strategy for said database query, said first search strategy being  
9       dependent on a first processor resource assignment at the time said step of constructing a first  
10      search strategy is performed;  
11              invoking said database query for execution in a first logical partition, said invoking step  
12       being performed after said step of constructing a first search strategy;  
13              comparing a second processor resource assignment to said first processor resource  
14       assignment, said second processor resource assignment being associated with said first logical  
15       partition at the time said invoking said database query for execution step is performed;  
16              determining whether a user has specified that automatic construction of another search  
17       strategy be disabled;  
18              automatically constructing a second search strategy dependent on said second processor  
19       resource assignment, said step of automatically constructing a second search strategy being  
20       performed dependent on the results of said comparing step, wherein said step of automatically  
21       constructing a second search strategy dependent on said second processor resource assignment is  
22       performed only if said determining step determines that a user has not specified that automatic  
23       construction of another search strategy be disabled.

7. (Cancelled)

1       8. (Currently Amended) The program product for database query optimization of ~~claim 7~~  
2       claim 11, wherein said respective processor resource assignment of each partition comprises a  
3       respective number of virtual processors of each partition, said respective number being an integer.

1       9. (Currently Amended) The program product for database query optimization of ~~claim 7~~  
2       claim 11, wherein said instructions further cause said computer system to perform the step of:  
3              saving said first search strategy in a persistent object for later execution, said saving step  
4       including saving said first processor resource assignment in said object.

1       10. (Previously Presented) A program product for database query optimization in a computer  
2       system having a plurality of central processors and a dynamic logical partitioning mechanism,  
3       said dynamic logical partitioning mechanism supporting a plurality of defined logical partitions of  
4       said computer system, each logical partition having a respective processor resource assignment,  
5       wherein each task executing in said computer system is assigned to a respective one of said  
6       logical partitions and wherein the definition of a plurality of logical partitions may be dynamically  
7       altered, said program product comprising a plurality of processor executable instructions recorded  
8       on signal-bearing media, wherein said instructions, when executed by at least one central  
9       processor of said computer system, cause said computer system to perform the steps of:  
10              receiving a definition of a database query;  
11              constructing a first search strategy for said database query, said first search strategy being  
12       dependent on a first processor resource assignment at the time said step of constructing a first  
13       search strategy is performed;  
14              invoking said database query for execution in a first logical partition, said invoking step  
15       being performed after said step of constructing a first search strategy;

16        comparing a second processor resource assignment to said first processor resource  
17        assignment, said second processor resource assignment being associated with said first logical  
18        partition at the time said invoking said database query for execution step is performed;

19        automatically constructing a second search strategy dependent on said second processor  
20        resource assignment, said step of automatically constructing a second search strategy being  
21        performed dependent on the results of said comparing step;

22        saving said first search strategy in a persistent object for later execution, said saving step  
23        including saving said first processor resource assignment in said object;

24        invoking a previously saved search strategy for execution in a second logical partition, said  
25        second logical partition being different from said first logical partition;

26        identifying a third processor resource assignment associated with said second logical  
27        partition;

28        comparing said third processor resource assignment to said first processor resource  
29        assignment; and

30        automatically constructing a third search strategy for execution of said database query  
31        depending on the results of said comparing step.

1        11. (Previously Presented) A program product for database query optimization in a computer  
2        system having a plurality of central processors and a dynamic logical partitioning mechanism,  
3        said dynamic logical partitioning mechanism supporting a plurality of defined logical partitions of  
4        said computer system, each logical partition having a respective processor resource assignment,  
5        wherein each task executing in said computer system is assigned to a respective one of said  
6        logical partitions and wherein the definition of a plurality of logical partitions may be dynamically  
7        altered, said program product comprising a plurality of processor executable instructions recorded  
8        on signal-bearing media, wherein said instructions, when executed by at least one central  
9        processor of said computer system, cause said computer system to perform the steps of:  
10              receiving a definition of a database query;  
11              constructing a first search strategy for said database query, said first search strategy being  
12        dependent on a first processor resource assignment at the time said step of constructing a first  
13        search strategy is performed;  
14              invoking said database query for execution in a first logical partition, said invoking step  
15        being performed after said step of constructing a first search strategy;  
16              comparing a second processor resource assignment to said first processor resource  
17        assignment, said second processor resource assignment being associated with said first logical  
18        partition at the time said invoking said database query for execution step is performed;  
19              determining whether a user has specified that automatic construction of another search  
20        strategy be disabled;  
21              automatically constructing a second search strategy dependent on said second processor  
22        resource assignment, said step of automatically constructing a second search strategy being  
23        performed dependent on the results of said comparing step, wherein said step of automatically  
24        constructing a second search strategy dependent on said second processor resource assignment is  
25        performed only if said determining step determines that a user has not specified that automatic  
26        construction of another search strategy be disabled.

27     12. (Previously Presented) A computer system, comprising:  
28         a plurality of central processing units;  
29         a memory;  
30         a logical partitioning mechanism supporting a plurality of defined logical partitions of said  
31         computer system, each logical partition having a respective processor resource assignment,  
32         wherein each task executing in said computer system is assigned to a respective one of said  
33         logical partitions and wherein the definition of said logical partitions may be dynamically altered;  
34         a database;  
35         a database management system for managing said database, wherein said database  
36         management system:  
37                 (a) performs query optimization of a database query for said database to produce a  
38                 first search strategy, said first search strategy being dependent on a first processor resource  
39                 assignment;  
40                 (b) responsive to invoking said first query search strategy for execution, compares  
41                 said first processor resource assignment with a second processor resource assignment  
42                 associated with a logical partition of execution at the time said first search strategy is  
43                 invoked for execution; and  
44                 (c) depending on the results of said comparison performed in (b), automatically  
45                 constructs a second search strategy dependent on said second processor resource  
46                 assignment;  
47         wherein said database management system further determines whether a user has specified that (c)  
48         be disabled, and disables (c) responsive to determining that a user has so specified.

1     13. (Original) The computer system of claim 12, wherein said respective processor resource  
2         assignment of each partition comprises a respective number of virtual processors of each partition,  
3         said respective number being an integer.

1    14. (Original) The computer system of claim 13, wherein said logical partitioning mechanism  
2    supports the definition of at least one set of processors which is shared by a set of said logical  
3    partitions, said set of said logical partitions containing at least two partitions, said respective  
4    processor resource assignment of each partition of said set of partitions including said set of  
5    processors.

1    15. (Original) The computer system of claim 12, wherein said database management system  
2    saves said first search strategy in a persistent object for later execution, said persistent object  
3    including said first processor resource assignment.

16. (Cancelled)